

# Iterativna lokalna pretraga

Aleksa Vošić, Lazar Perišić,  
Anđela Križan, Anđela Janošević

Matematički fakultet  
Univerzitet u Beogradu

Beograd, 2020.

# Uvod

- Metaheurističke metode za rešavanje teških optimizacionih problema
- Prilikom dizajniranja metaheuristike, poželjno je da bude jednostavna, efikasna, opšte namene
- Idealan slučaj je kada se metaheuristika može koristiti bez ikakvog znanja o zavisnosti od problema
- Znanje specifično za problem mora biti inkorporirano u metaheuristiku da bi se dostiglo vrhunsko stanje
- Pokušavamo da dekomponujemo metaheuristički algoritam na nekoliko delova:
  - potpuno opšti namenski deo
  - svako znanje specifično za problem ugrađeno u metaheuristiku bilo bi odvojeno u drugi deo

## Nešto više

- Iterativna lokalna pretraga pruža jednostavan način da se zadovolje svi ovi zahtevi
- Suština iterativne lokalne pretrage je da se izbegne zaglavljivanje u lokalnom minimumu tako što u više iteracija primenjuje lokalnu pretragu na novo generisano početno rešenje
- Ova ideja ima dugu istoriju, a njeno ponovno otkriće od strane mnogih autora dovelo je do mnogo različitih imena za iterativnu lokalnu pretragu poput iterativnog spusta, Markovljevi lanci velikog koraka, iterativni Lin-Kernigan, lančana lokalna optimizacija...

# Ideja iza iterativne lokalne pretrage

# Implementacija iterativne lokalne pretrage

# Početno rešenje i Perturbacija

# Kriterijum prihvatanja i Lokalna pretraga

# Primene iterativne lokalne pretrage



# Problem trgovačkog putnika

## Problemi raspoređivanja

ILS se takođe može uspešno primeniti i na probleme raspoređivanja kao što su:

- **problem rasporeda na jednoj mašini pomoću ukupnog kašnjenja sa težinskim koeficijentima** (*eng.* single machine total weighted tardiness problem) tj. SMTWTP
- **problem rasporeda  $n$  proizvoda na  $m$  mašina** (*eng.* flow shop problem) tj. FSP
- **problem raspoređivanja poslova** (*eng.* job shop scheduling problem) tj. JSSP

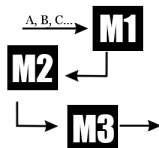
kao i mnogi drugi.

## Single machine total weighted tardiness problem

Problem se sastoji od skupa nezavisnih poslova sa različitim vremenima obrade, težinama (cenama), kao i rokovima do kad poslovi moraju biti završeni predviđenim da se obrađuju na jednoj datoj mašini.

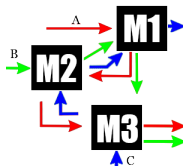
- Početno rešenje se može generisati pomoću raznih heuristika konstrukcija (EDD, MDD, AU).
- Perturbacija se sastoji od niza proizvoljnih nezavisnih poteza razmene: posao na poziciji  $i$  se razmenjuje sa poslom na poziciji  $j$ . Razmena  $i, j$  se ne preklapa sa razmenom  $k, l$  ukoliko važi:  $\min\{i, j\} \geq \max\{k, l\}$  ili obrnuto.
- Kriterijum prihvatanja: backtrack korak, svako novo najbolje lokalno stanje se prihvata, algoritam se restartuje sa najboljim pronađenim rešenjem.

## Flow shop problem



- Početno stanje se generiše uz pomoć NEH heuristike.
- Perturbacija se generiše pravljenjem pomoću dva različita tipa poteza:
  - $\pi = (\pi(1), \dots, \pi(i), \pi(i+1), \dots, \pi(n)) \rightarrow \pi' = (\pi(1), \dots, \pi(i+1), \pi(i), \dots, \pi(n))$
  - $\pi = (\pi(1), \dots, \pi(i), \dots, \pi(j), \dots, \pi(n)) \rightarrow \pi' = (\pi(1), \dots, \pi(j), \dots, \pi(i), \dots, \pi(n))$
- Kriterijum prihvatanja: može se uvek birati stanje koje je bolje i da se ono zadrži, ili napredniji slučaj koji analizira novo stanje i prihvata ga korišćenjem odgovarajuće verovatnoće.

# Job shop scheduling problem



Postoje razni pristupi ovom problemu, opisaćemo pristup koji je definisala H.R. Lourenço.

- Početna rešenja imaju veoma mali uticaj na sam algoritam.
- Formiranja perturbacije je zasnovano na definisanju jedno-mašinskog ili dvo-mašinskog potproblema fiksiranjem broja promenljivih u trenutnom rešenju i rešavanjem ovih potproblema nekom od heuristika ili nekim polinomijalnim algoritmima

## Efektivnost i efikasnost ILS algoritma

% Vremenskog uvećanja u odnosu na optimalno rešenje =

$$\frac{Alg_{rez} - Opt_{rez}}{Opt_{rez}} \cdot 100$$

Problem	SAOP	SPIRIT	GAChen	GAMIT	ILS
20x5	1.39 ( $\leq 0.5$ )	5.22 ( $\leq 0.5$ )	3.82 ( $\leq 0.5$ )	4.21 ( $\leq 0.5$ )	0.24 (4.01)
20x10	2.66 ( $\leq 0.5$ )	5.86 ( $\leq 0.5$ )	4.89 ( $\leq 0.5$ )	5.40 ( $\leq 0.5$ )	0.77 (4.09)
20x20	2.31 ( $\leq 0.5$ )	4.58 ( $\leq 0.5$ )	4.17 (0.60)	4.53 ( $\leq 0.5$ )	0.85 (4.63)
50x5	0.69 ( $\leq 0.5$ )	2.03 ( $\leq 0.5$ )	2.09 (0.77)	3.11 ( $\leq 0.5$ )	0.12 (6.38)
50x10	4.25 (0.60)	5.88 (0.52)	6.60 (1.00)	8.38 (0.52)	2.01 (9.94)
50x20	5.13 (1.04)	7.21 (0.97)	8.03 (1.45)	10.65 (0.96)	3.29 (11.82)
100x5	0.40 (0.60)	1.06 (0.53)	1.32 (1.79)	5.41 (0.52)	0.11 (15.31)
100x10	1.88 (1.10)	5.07 (1.03)	3.75 (2.26)	12.05 (1.02)	0.66 (18.79)
100x20	5.21 (2.09)	10.15 (2.00)	7.94 (3.24)	18.24 (1.99)	3.17 (24.04)
200x10	1.56 (2.29)	9.03 (2.25)	2.70 (5.97)	7.52 (2.20)	0.49 (33.73)
200x20	4.83 (4.59)	16.17 (4.51)	7.07 (8.18)	15.35 (4.50)	2.74 (41.80)
500x20	3.40 (39.48)	13.57 (39.70)	4.61 (55.30)	12.17 (37.82)	1.29 (192.03)
Average	2.81 (4.42)	7.15 (4.38)	4.75 (6.77)	8.92 (4.21)	1.31 (30.55)

## Zaključak

- ILS poseduje mnoge poželjne karakteristike metaheuristike: jednostavan je, lagan za implementaciju, robustan i veoma efikasan
- Suštinska ideja ILS-a leži u fokusiranju pretraživanja ne na celokupnom prostoru rešenja, već na manjem potprostoru koji je definisan rešenjima koja su lokalno optimalna za datu optimizaciju
- Koliko će se ovaj pristup pokazati efikasnim, uglavnom zavisi od izbora lokalne pretrage, perturbacija i kriterijuma prihvatanja
- Zbog svojih karakteristika verujemo da je ILS obećavajući i moćan algoritam za rešavanje stvarnih kompleksnih problema

# Literatura